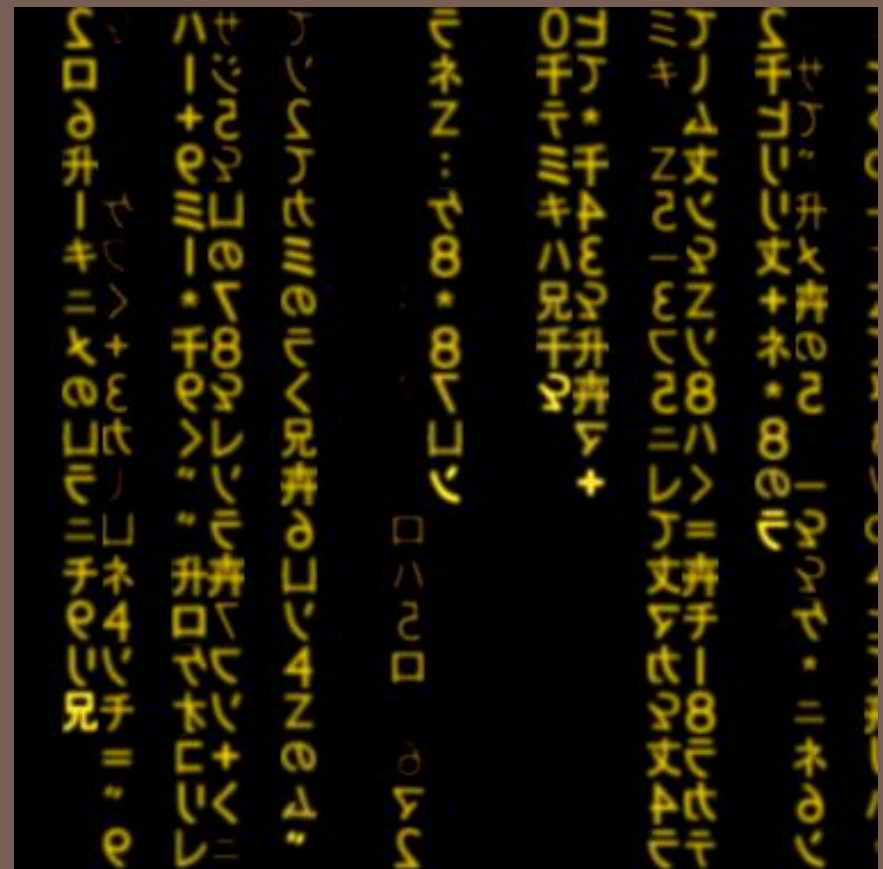


Αντώνης Χρυσόπουλος  
Στέλιος Μόσχογλου  
Θεοδόσης Σουργκούνης



## ΔΟΜΗΣΗ ΚΩΔΙΚΑ

Μαθαίνω παίζοντας

# Σήμερα παιδάκια μου θα μάθουμε...

1. Δόμηση Κώδικα
2. Στυλ Κώδικα
3. Συνεργασία Σε Ομάδες - Διαχωρισμός Σε Αρχεία
4. Front-End – Back-End
5. Το Πρότυπο MVC

# Παράδοση

---

## 1. Δόμηση Κώδικα

# Δόμηση Κώδικα

- Καθαρότερος κώδικας **για εμάς**.
  - ▣ Επεξεργασία μετά από **ένα μήνα** ή ένα χρόνο.
  - ▣ Πρέπει να τον ξανακαταλάβουμε.
- Καθαρότερος κώδικας για τους άλλους.
  - ▣ Κι άλλοι διαβάζουν τον κώδικά σας.
  - ▣ Πρέπει να τον καταλάβουν.
  - ▣ Πρέπει να τον αλλάξουν.

Αυτό σημαίνει ότι θα πρέπει να δώσουμε την αρμόζουσα βαρύτητα στη δόμησή του.

# Δόμηση Κώδικα (2)

- Πλεονεκτήματα ορθής δόμησης: Maintainability.
  - ▣ Ο κώδικας μπορεί να αλλάξει εύκολα.
  - ▣ Είναι προσαρμόσιμος.
  - ▣ Είναι επεκτάσιμος.
  - ▣ Είναι ευανάγνωστος.
  - ▣ Εκφράζει την σκέψη μας.

“Programs must be written for **people** to read, and only incidentally for machines to execute.”

- Abelson & Sussman

Structure and Interpretation of Computer Programs

# Πες μας Κωστάκη την απάντηση.

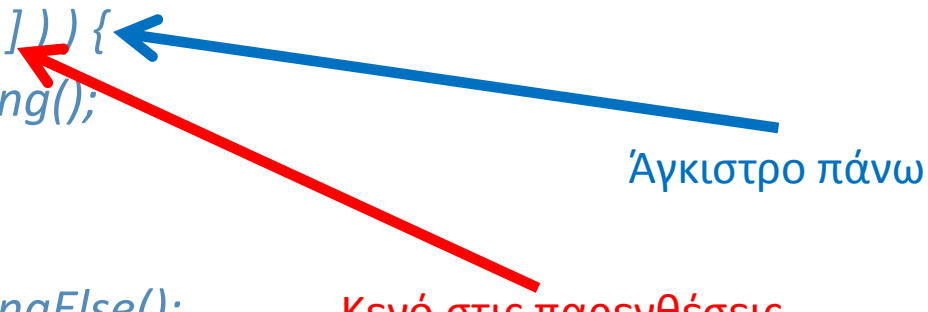
1. Δόμηση Κώδικα
2. Στυλ Κώδικα

# Στυλ Κώδικα

- Το στυλ κώδικα ορίζει την μορφή του κώδικα
  - ▣ πού έχει κενά και πού όχι.
  - ▣ πόσα κενά έχει.
  - ▣ πότε γίνεται indentation.
  - ▣ πότε γίνονται αλλαγές γραμμών κ.α.
- Δεν έχει σημασία ποιο στυλ χρησιμοποιείτε, δεν έχει σημασία αν έχετε πολλά κενά ή λίγα.
- **Διαλέξτε** ένα στυλ και **κρατήστε** το
  - ▣ Μία ομάδα → Ένα στυλ.
  - ▣ Ένα λογισμικό → Ένα στυλ.
- **Συνέπεια!** Οτιδήποτε άλλο είναι αντιπαραγωγικό.

# Παράδειγμα

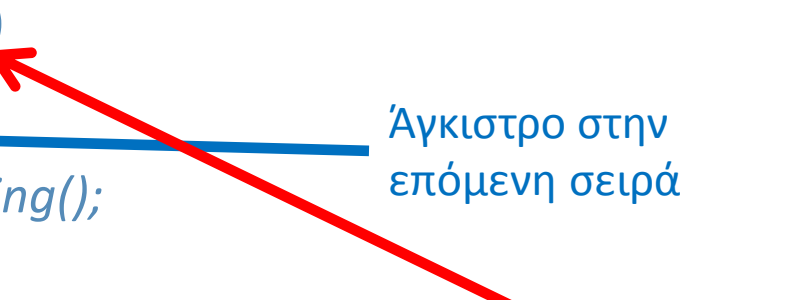
```
if ( isset( $foo[ 'bar' ] ) ) {  
    doSomething();  
}  
else {  
    doSomethingElse();  
}
```



Άγκιστρο πάνω

Κενό στις παρενθέσεις

```
if(isset($Foo['bar']))  
{  
    DoSomething();  
}  
else  
{  
    DoSomethingElse();  
}
```



Άγκιστρο στην επόμενη σειρά

Χωρίς κενό στις παρενθέσεις



# Ασυνέπεια

```
if ( $a ) {  
    doSomething();  
}  
else { doSomethingElse(); }
```

ΑΝΤΕ ΠΑΟΚΑΡΑ ΟΤΙ ΝΑ ΝΑΙ!!!



# Στυλ Κώδικα

- Αν διαφωνείτε στο στυλ...
  - ▣ Πιο σημαντικό να γράφετε **στο ίδιο στυλ** με τους συναδέλφους σας.
  - ▣ Παρά να γράφετε σε διαφορετικά στυλ στην ίδια ομάδα.
- Σε ένα ήδη υπάρχον project, χρησιμοποίησε το στυλ που υπάρχει ήδη (ΠΡΟΦΑΝΩΣ!!!).
- Υπάρχουν έτοιμες προτάσεις
  - ▣ Mozilla.
  - ▣ Google.
  - ▣ και άλλες. Αναζητήστε για “Coding Style”.
- Βρείτε μία που σας ταιριάζει.

# Μαιρούλα σταμάτα να μιλάς

1. Δόμηση Κώδικα
2. Στυλ Κώδικα
3. Συνεργασία Σε Ομάδες - Διαχωρισμός Σε Αρχεία
4. Front-End – Back-End

# Μόνος σου, δεν είσαι ΤΙΠΟΤΑ!!!

- Facebook:
  - ▣ **4** ιδρυτές
  - ▣ 1700 άτομα σήμερα.
- Google:
  - ▣ **2** ιδρυτές
  - ▣ 23000 άτομα σήμερα.
- Microsoft: **2** ιδρυτές.
- Apple: **3** ιδρυτές.
- YouTube: **3** ιδρυτές.

Άρα μόνος σου δεν πάς πουθενά, πρέπει να μάθεις να συνεργάζεσαι.

# Συνεργασία σε Ομάδες

- Απαιτείται καταμερισμός εργασιών.
- Αυτό σημαίνει **σωστή δόμηση**.
- Διαχωρισμός σε αρχεία.
- Όλοι πρέπει να συμφωνήσουμε:
  - ▣ Τι είδους κώδικα γράφουμε πού;
- Πού θα γραφεί ο κώδικας...
  - ▣ PHP, για την σύνδεση στη βάση δεδομένων;
  - ▣ HTML, για την φόρμα δημιουργίας λογαριασμού;
  - ▣ SQL, για την δημιουργία λογαριασμού;
- **Κάθε** πράγμα πρέπει να έχει **μία** θέση.

# Διαχωρισμός σε Αρχεία

- Πρώτη ιδέα:
  - ▣ Αντί να έχω ένα μεγάλο αρχείο, έχω πολλά μικρότερα.
  - ▣ Ευκολότερη συνεργασία.
  - ▣ Ο καθένας δουλεύει στα αρχεία που τον αφορούν.
  
- Δεύτερη ιδέα
  - ▣ Συναρτήσεις.
  - ▣ Επαναχρησιμοποίηση κώδικα.
  - ▣ Συχνά μπορούν να χρησιμοποιηθούν σε αρχεία.

# Front-End – Back-End

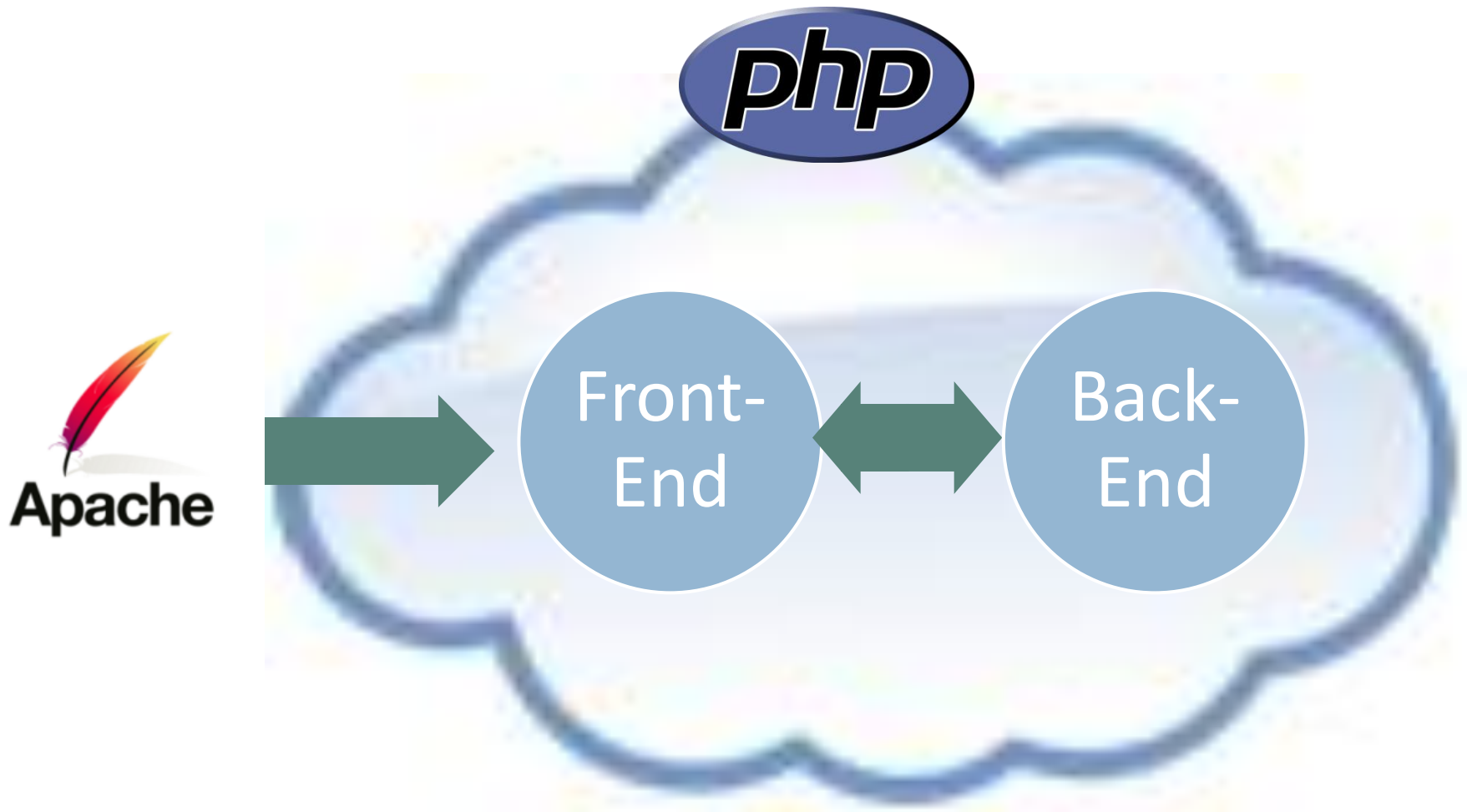
## □ Front-end:

- ▣ Κώδικας που αναφέρεται στην διεπαφή χρήστη
- ▣ PHP που παράγει άμεσα HTML
- ▣ Στατικό HTML
- ▣ CSS
- ▣ Javascript
- ▣ Εικόνες κ.α.

## □ Back-end:

- ▣ Κώδικας που επεξεργάζεται τα δεδομένα
- ▣ PHP που δεν παράγει HTML
- ▣ Συναρτήσεις επεξεργασίας δεδομένων
- ▣ Συναρτήσεις αποθήκευσης δεδομένων
- ▣ SQL

# Front-End – Back-End (2)





# Front-End – Back-End (3)

- Χωρίζουμε σε ξεχωριστά αρχεία
  - ▣ To front-end.
  - ▣ To back-end.
- Στο front-end δεν υπάρχει SQL.
- Στο back-end δεν υπάρχει HTML/CSS.
- Το **front-end** περιγράφει την παραγωγή της **διεπαφής χρήστη**.
- Το **back-end** περιγράφει την **επεξεργασία, αποθήκευση, ανάκτηση** δεδομένων.
- Το back-end συχνά αναφέρεται και ως **models**
  - ▣ Συχνό φαινόμενο σε web εφαρμογές:
    - **Φάκελος models** που περιέχει τον back-end κώδικα

# Decoupling

- Κάθε τμήμα κώδικα γνωρίζει μόνο όσα **χρειάζεται** να γνωρίζει.
- Κάθε είδους «γνώση» υπάρχει μόνο σε **ένα** σημείο του κώδικά μας.
- Front-end γνωρίζει:
  - ▣ Ότι χρησιμοποιούμε HTML.
  - ▣ Ποια έκδοση της HTML χρησιμοποιούμε.
- Back-end γνωρίζει:
  - ▣ Ότι χρησιμοποιούμε MySQL.
  - ▣ Ποια έκδοση της MySQL χρησιμοποιούμε.
  - ▣ Ποιο είναι το σχήμα μας.

# Decoupling (2)



- Front-end **δεν** γνωρίζει:
  - ▣ Αν χρησιμοποιούμε **αρχεία** ή **βάση δεδομένων** για αποθήκευση.
  - ▣ Αν αποθηκεύω το όνοματεπώνυμο ως όνομα + επώνυμο.
  - ▣ Αν τα δεδομένα προέρχονται από ανάκτηση ή από υπολογισμό.
  
- Back-end **δεν** γνωρίζει:
  - ▣ Ότι παράγουμε HTML.
  - ▣ Αν χρησιμοποιούμε XHTML 1.0 Strict ή όχι.
  - ▣ Ότι ο χρήστης είναι άνθρωπος και όχι η Sarah Jessica Parker (που είναι άλογο).

# Παράδειγμα

<h2>Τι είπαν οι άλλοι??</h2>

<ol>

<?php

\$res = mysql\_query(

"SELECT

username, text

FROM

shouts INNER JOIN users

ON shouts.userid = users.userid

ORDER BY

created DESC;"

);

while ( \$row = mysql\_fetch\_array( \$res ) ) {

?> <li><strong> <?php

echo \$row[ 'username' ];

?> :</strong>

<span> <?php

echo \$row[ 'text' ];

?> </span></li> <?php

}

?>

</ol>

Front-End

Back-End

Front-End

# Πως Χωρίζουμε τον Κώδικα??

- Το HTML στο front-end.
- Η SQL στο back-end.
- Σε μία ομάδα ο κάθε προγραμματιστής επιλέγει αν θα ασχοληθεί με το back-end ή το front-end.
  - ▣ Ταχύτερη ανάπτυξη.
  - ▣ Δεν γράφεται το ίδιο πράγμα 2 φορές.

# SQL Ερώτημα

models/shouts.php:

```
$res = mysql_query(  
    "SELECT  
        username, text  
    FROM  
        shouts CROSS JOIN users  
    ON shouts.userid = users.userid  
    ORDER BY  
        created DESC;"  
);
```

Back-End



# Υπόλοιπος Κώδικας

```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
include 'models/shouts.php';
```

```
while ( $row = mysql_fetch_array( $res ) ) {
```

```
?> <li><strong> <?php
```

```
echo $row[ 'username' ];
```

```
?> :</strong>
```

```
<span> <?php
```

```
echo $row[ 'text' ];
```

```
?> </span></li> <?php
```

```
}
```

```
?>
```

```
</ol>
```

Αδιαφανής Αναφορά

Back-End

Front-End ?

# Αλλαγή Διεπαφής

```
<h2>Τι είπαν οι άλλοι</h2>
```

```
<ol>
```

```
<?php
```

```
include 'models/shouts.php';
```

```
$shouts = GetShouts();
```

Διεπαφή (API) Front-End/ Back-End

```
foreach ( $shouts as $shout ) {
```

```
    ?> <li><strong> <?php
```

```
    echo $shout[ 'username' ];
```

```
    ?> :</strong>
```

```
    <span> <?php
```

```
    echo $ shout[ 'text' ];
```

```
    ?> </span></li> <?php
```

```
}
```

```
?>
```

```
</ol>
```



# models/shouts.php

```
function GetShouts() {  
    $rows = array();  
    $res = mysql_query(  
        "SELECT username, text  
        FROM  
            shouts CROSS JOIN users  
        ON shouts.userid = users.userid  
        ORDER BY created DESC;"  
    );  
    while ( $row = mysql_fetch_array( $res ) ) {  
        $rows[] = array(  
            $row[ 'username' ], $row[ 'text' ]  
        );  
    }  
    return $rows;  
}
```

# Front-End – Back-End

## □ Front-end:

- Εμπεριέχει όλη την **λογική** διεπαφής χρήστη
- Παραγωγή HTML
- CSS/JS
- **Δεν** γνωρίζει για τον τρόπο αποθήκευσης των δεδομένων
- **Δεν** γνωρίζει πώς ανακτούνται τα δεδομένα

## □ Back-end που ζει στον φάκελο “models”:

- Εμπεριέχει όλη την **λογική** επικοινωνίας με την βάση
- Αποθήκευση
- Ανάκτηση
- Επεξεργασία
- **Δεν** γνωρίζει για τον τρόπο χρήσης των δεδομένων
- **Δεν** γνωρίζει από πού προήλθαν τα δεδομένα

## Παράδειγμα 2

```
if ( isset( $_SESSION[ 'username' ] )
```

```
&& isset( $_POST[ 'shout' ] ) ) {
```

```
    $shout = $_POST[ 'shout' ];
```



Front-End

```
    mysql_query(
```

```
        "INSERT INTO
```

```
        shouts
```

```
        SET
```

```
        text = ' ' . $shout . ' ',
```



Back-End

```
        userid = ' ' . $_SESSION[ 'userid' ] . ',
```

```
        created = NOW();"
```

```
    );
```

```
    header( 'Location: index.php' );
```

```
}
```

```
else {
```

```
    ?>Πρέπει να έχεις κάνει είσοδο.<?php
```



Front-End

```
}
```

# Back-End (models/shouts.php)

```
function SaveShout( $userid, $text ) {  
    mysql_query(  
        "INSERT INTO  
        shouts  
        SET  
        text = '$text',  
        userid = $userid,  
        created = NOW();" );  
}
```

# Front-End

```
if ( isset( $_SESSION[ 'username' ] )
    && isset( $_POST[ 'shout' ] ) ) {
    include 'models/shouts.php';

    SaveShout(
        $_SESSION[ 'userid' ],
        $_POST[ 'shout' ]
    );

    header( 'Location: index.php' );
}
else {
    ?>Πρέπει να έχεις κάνει είσοδο.<?php
}
```

# Just Testing!!! What is...

- Εμφάνιση μηνύματος σφάλματος (Δεν έχεις κάνει login).
- Ανάγνωση δεδομένων GET.
- Ανάγνωση δεδομένων POST.
- Προσθήκη εγγραφής στη βάση δεδομένων.
- Εμφάνιση κουμπιού που αποθηκεύει.
- Χρήση `mysql_fetch_array` για ανάγνωση δεδομένων.
- Εύρεση μέσου όρου βαθμολογιών μαθητών.
- Χρωματισμός γραμμάτων διεπαφής με κόκκινο χρώμα.

# Τέρμα το διάλλειμα, τα κεφάλια μέσα...

1. Δόμηση Κώδικα
2. Στυλ Κώδικα
3. Συνεργασία Σε Ομάδες - Διαχωρισμός Σε Αρχεία
4. Front-End – Back-End
5. Το Πρότυπο MVC

# Μοντέλο MVC

- MVC = Model-View-Controller.
- Καλός ο διαχωρισμός Front-end / Back-end.
- Back-end = Model.
- Όμως στο Front-end μας μπλέκουμε δύο πράγματα.
  - ▣ Την παραγωγή της διεπαφής (View).
  - ▣ Την συλλογή δεδομένων και τις απαραίτητες κλήσεις (Controller).



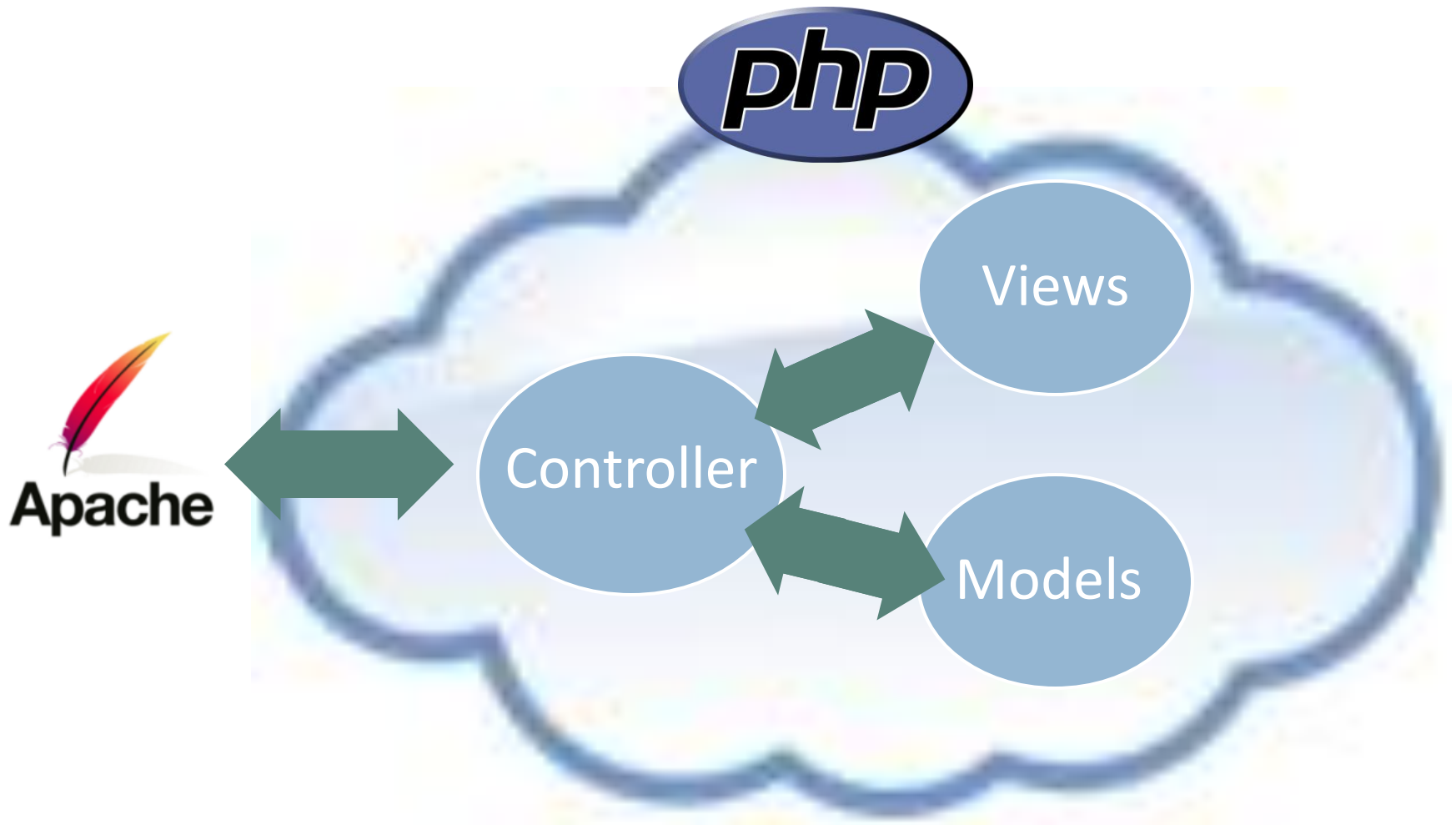
# Views

- Αποκλειστικά υπεύθυνα για την παραγωγή διεπαφής.
- Κυρίως HTML κώδικας.
- Μόνο η απαραίτητη PHP.
  - ▣ π.χ. `foreach` για παραγωγή επαναλαμβανόμενων τμημάτων.
- Πρόσβαση σε συγκεκριμένες ονομασμένες μεταβλητές.
- Λέγονται και «**Templates**» ή πρότυπα.

# Controllers

- Αποφασίζουν ποιο view και ποιο model θα κληθεί.
  - Παίρνουν τα δεδομένα από τον χρήστη.
  - Δίνουν τα **δεδομένα του χρήστη** στο σωστό model.
  - Δίνουν τα δεδομένα του χρήστη **και του model** στο view.
  - Στέλνουν πίσω τα δεδομένα στον χρήστη.
- 
- Το «λεπτότερο» τμήμα.
  - Το πρώτο και τελευταίο πράγματα που τρέχει.
  - Έχει τον **έλεγχο** (controller = ελεγκτής).

# MVC Schema



# Front-End (2)

```
if ( isset( $_SESSION[ 'username' ] )  
&& isset( $_POST[ 'shout' ] ) ) {  
    include 'models/shouts.php';
```

Controller

```
SaveShout(  
    $_SESSION[ 'userid' ],  
    $_POST[ 'shout' ]  
);
```

Controller

```
header( 'Location: index.php' );  
}
```

Controller

```
else {  
    ?>Πρέπει να έχεις κάνει είσοδο.<?php  
}
```

View

```
if ( isset( $_GET[ 'username' ] ) ) {
    $res = mysql_query(
        "SELECT userid
        FROM users
        WHERE
            username = " . $_GET[ 'username' ] . "
        LIMIT 1;" )
    );
    if ( mysql_num_rows( $res ) == 1 ) {
        ?>Username already taken :-(
        <input type="text" value="" /> <?php
    }
    else {
        ?> <input type="text" value=" <?php
        echo $_GET[ 'username' ];
        ?> " /><?php
    }
}
else {
    ?> <input type="text" value="" /> <?php
}
```

Controller

Model

Controller

Model

View

# 1. Controller

```
$exists = false;  
$username = "";  
if ( isset( $_GET[ 'username' ] ) ) {  
    include 'models/user.php';  
    $exists = UsernameExists( $_GET[ 'username' ] );  
    $username = $_GET[ 'username' ];  
}  
include 'views/register.php';
```

## 2. Model

```
function UsernameExists( $username ) {  
    $res = mysql_query(  
        "SELECT  
        userid  
        FROM  
        users  
        WHERE  
        username = '$username'  
        LIMIT 1;"  
    );  
    return mysql_num_rows( $res ) == 1;  
}
```

# 3. View

```
<?php
if ( $exists ) {
    ?> Το όνομα χρήστη υπάρχει ήδη :-( <?php
}
?>

<input type="text" value="<?php
    echo $username;
?> " />
```



# Που ανηκούν τα παρακάτω στο MVC?

- Εμφάνιση μηνύματος σφάλματος (Δεν έχεις κάνει login)
- Ανάγνωση δεδομένων GET
- Ανάγνωση δεδομένων POST
- Προσθήκη εγγραφής στη βάση δεδομένων
- Εμφάνιση κουμπιού που αποθηκεύει
- Χρήση `mysql_fetch_array` για ανάγνωση δεδομένων
- Εύρεση μέσου όρου βαθμολογιών μαθητών
- Χρωματισμός γραμμάτων διεπαφής με κόκκινο χρώμα
- Include του view
- Include του model

# In Conclusion



# ΜΠΡΑΒΟ!!!

- Είσαι η καλύτερη!!!

